Proceedings of the 10th International Conference on Human-Computer Interaction, Crete, June 22-27, 2003.

ANALYSIS AND VERIFICATION OF HUMAN-AUTOMATION INTERFACES

Asaf Degani
NASA Ames Research Center
Moffett Field, CA

Michael Heymann
Technion - Israel Institute of Technology
Haifa, Israel

ABSTRACT

This paper addresses the problem of verifying and designing human-automation interfaces. The approach focuses on what information is provided to the user (and not on how this information is presented). We describe a formal methodology for verification of interfaces. The methodology is aimed at proving that the information provided to the user via the display (e.g., modes and parameters), enables the user to perform his or her tasks successfully. We assert that a display and corresponding user-manuals are correct if there exist no *error states*, no *blocking states*, and no *augmenting states*. The essentials of the methodology, which can be automated and applied to the verification of large and complex systems, are discussed and illustrated via a simplified automotive example. An extension of this formal approach for generating interfaces and associated user-manuals is briefly discussed.

INTRODUCTION

Automated control systems such as automotive, medical equipment, and avionics exhibit extremely complex behaviors. These large systems react to external events and internal events, as well as user-initiated events. For the user to be able to monitor the machine and interact with it to achieve a task, the information provided to the user about the machine must, above all, be correct. In principle, correct interaction can always be achieved by providing the user with the full detail of the underlying machine behavior, but in reality, the sheer amount of such detail is generally impossible for the user to absorb and comprehend. Therefore, the machine's interface and related user-manuals are always a reduced, or abstracted, description of the underlying machine behavior. Naturally, we all prefer interfaces that are also simple and straightforward. This, of course, reduces the size of user manuals, training costs, and perceptual and cognitive burdens on the user (Abbott, Slotte, & Stimson, 1996).

In this paper, we will briefly present an approach and methodology for verifying interfaces and user manuals. The methodology evaluates whether the interface and user-manual information are correct and free of errors, given a description of the machine, the user's task, and the interface (Degani & Heymann, 2002).

FORMAL ASPECTS OF HUMAN-MACHINE INTERACTION

In analyzing human automation interaction from a formal perspective, we consider here three major elements: (1) the behavior of the machine (in terms of states and transitions), (2) the user's tasks, and (3) the interface.

Machine

Computer-based system and automated control systems can be described using a variety of formal models. In this paper, we model a machine as a finite system of states (but note that the methodology described here is general and can be applied to any formalism). Some of the transitions in the system are triggered by the user (e.g., a driver placing the automatic gear in Drive). Other transitions are automatic and are triggered either by the machine's internal dynamics (e.g., automatic transition from 1st to 2nd gear), or by the external environment (e.g., when the car's speed is above 180 miles per hour, the transmission prevents the driver from further speed increase). In the models described here, we depict user-triggered transitions by solid lines, while automatic transitions are broken lines. The transitions are labeled by Greek symbols indicating the events under which the machine moves from state to state.

The machine in Figure 1 describes a simplified three-speed transmission system of a vehicle. The transmission has eight states (representing internal torque-levels). These are grouped into three speed modes: LOW, MEDIUM, and HIGH. States L1, L2, and L3 are in the LOW speed mode; M1 and M2 in the MEDIUM speed mode; and H1, H2, H3 in HIGH. The transmission shifts up and down either automatically (based on throttle, engine, and speed values) or manually by pushing a lever (Figure 2). Manual up-shifts are denoted by event β and down-shifts by event ρ. Automatic up-shifts are denoted by event δ, and automatic down-shifts by event γ.

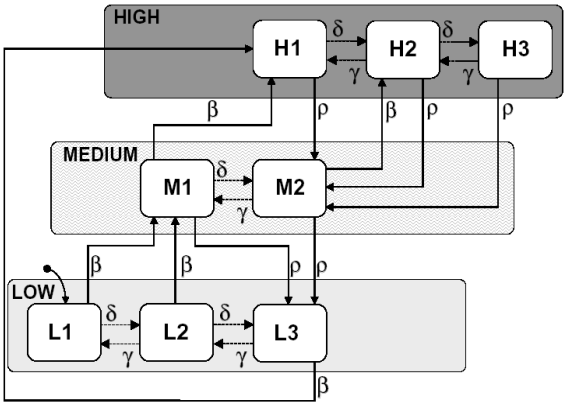

Figure 1. Transmission system

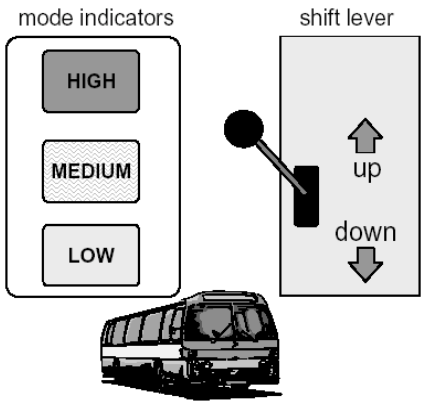

Figure 2. Display and control panel

User's Tasks

The second element is the user's tasks, which in case of this transmission system, consists of tracking the three speed modes unambiguously. In other words, the user must be able to determine the current mode of the machine and predict the next mode of the machine. This requirement is akin to the type of questions users usually ask about automated systems: "What's it doing now?" "What's it going to do next?" and "Why is it doing that?" (Wiener, E. L. personal communication, April 5, 2002). We describe the user's task by partitioning the machine's state-set (the eight internal states in Figure 1) into three disjoint regions: low, medium, and high. Note however that the user is required to track only the modes that correspond to these regions, and *not* every individual state of the machine.

Interface

As discussed earlier, the interface generally provides the user with a simplified view of the machine. In almost any display, especially those for automated systems, many of the machine's internal events and states are hidden from the user—otherwise, the size of cockpit displays, for example, would be colossal. Hence the display provides only partial, i.e., abstracted, information about the underlying behavior of the machine. Since we want an interface that is correct and succinct, the essence of the interface design problem centers on what information can be safely removed away, or abstracted, from the display and what information must be presented.

Figure 3 describes one proposed display for the transmission system. The display indicates the three primary modes (LOW, MEDIUM and HIGH), and the driver shifts among modes by pushing up or down on the gear lever. What is also being removed from the interface, user-manual, and consequently from the user's awareness is the automated internal transitions that take place within each mode, or gear. For example, the LOW mode has three possible internal states, L1, L2, and L3. When the user first up-shifts manually into low gear, L1 is the active state. When the driver increases speed, an automatic transition to L2 takes place. This internal transition is not evident to the driver, who is aware only of being in LOW mode.

EVALUATION OF INTERFACES

In addition to the display indications, Figure 3 also shows the manual transitions that are needed to move the transmission system from one mode to another (this information is provided in the user-manual). This proposed display is very simple and straightforward: it shows only the three modes (LOW, MEDIUM, and HIGH), all internal states are removed, and all the automatic transitions are suppressed.

Is this an adequate display?

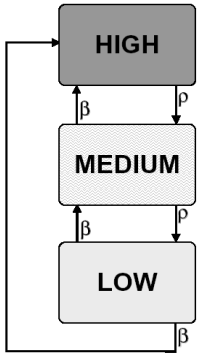

Figure 3. Proposed display.

Intuitively it looks fine, but let's look at it more carefully: The manual shifts from MEDIUM to HIGH or down to LOW, as well as the down-shift from HIGH to MEDIUM, are always predictable—the user will be able to anticipate the next mode of the machine. However, note that the transitions out of LOW depend on the internal states: up-shifts from L1 and L2 take us to MEDIUM, while the up-shift from L3 switches the transmission to HIGH. What we have here is that the same event (β) takes us to two different modes. But since the display hides from us which internal state we are in, we will not be able to predict if the system will transition to MEDIUM or HIGH. Therefore, we must conclude that this display is incorrect and inadequate for the task.

An alternate user model that may remedy the above problem is depicted in Figure 4. This modified display has two LOW modes (LOW-1 and LOW-2). The user manual further explains that the transitions between LOW-1 and LOW-2 occur automatically, and that upon up-shift from LOW-1, the system transitions to MEDIUM, while on up-shift from LOW-2, the system goes to HIGH.

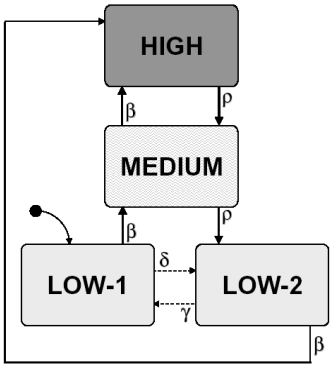

Figure 4. Alternate display

FORMAL VERIFICATION OF INTERFACES

Again, we ask: is this a good interface?

Well, by intuitive inspection it seems quite reasonable—we have taken care of the problem with the manual up-shift out of LOW. But let us try to verify this intuition in a more methodological way. The way we do this is by creating a composite model of the display of Figure 4 and the machine model. This is the model presented in Figure 5(c).

We evaluate the composite model by exploring all the composite states. The machine (Figure 5a) starts in state L1 and the display (Figure 5b) starts in LOW-1. So the first composite state is "L1, low-1." Upon an automatic up-shift transition (event δ), the machine transitions to L2 and the display to low-2, and now we are in composite state "L2, low-2." At this point, the user decides to up-shift manually. The machine will transition to state M1, yet according to Figure 5(b), we are now in HIGH mode. The new composite state is "M1, HIGH," which of course is a contradiction! The user thinks he is in HIGH mode (and the display confirms) where in fact the underlying machine is in MEDIUM (state M1). The resulting ambiguity is a classical mode error (Norman, 1983), and we call such a composite configuration an *error-state*.

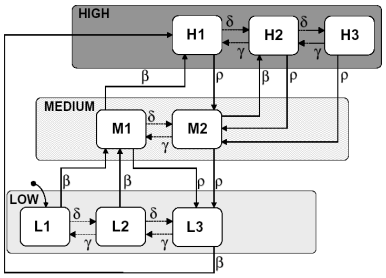

Figure 5a. Machine model

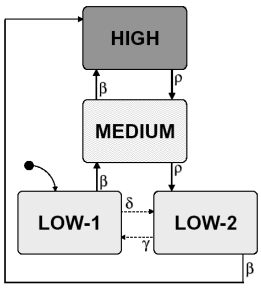

Figure 5b. Display model

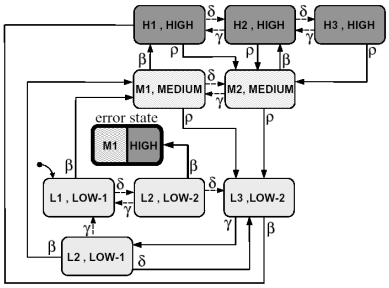

Figure 5c. Composite model

CONCLUSIONS

The objective of the verification methodology is to determine whether a given interface enables the user to operate the machine correctly. The essence of the verification procedure is to check whether the interface "marches" in synchronization with the machine model. We assert that a user-model is correct if there exist no *error states*, no *blocking states*, and no *augmenting states* in the composite model. An error state represents a divergence between the machine and its interface. That is, the display tells the user that the machine is in one mode when in fact the machine is in another. A blocking state represents a situation in which the machine can in fact trigger a transition from one mode to another, yet this information is not provided to the user (and when the transition happens, the user is surprised). An augmenting state is a situation in which the user is told that a certain mode change is possible, when in fact it may be the case that the machine will not switch into this mode or sub-mode. The details of this verification methodology and its application to an automated flight control system are provided elsewhere (Degani & Heymann, 2002).

Going back to the transmission system, it is possible to concoct other displays and then iteratively employ the verification procedure to determine their correctness. It turns out that there exist a composite model that exhibits no error states, no blocking states, and no augmenting states. The corresponding display is the one depicted in Figure 6.

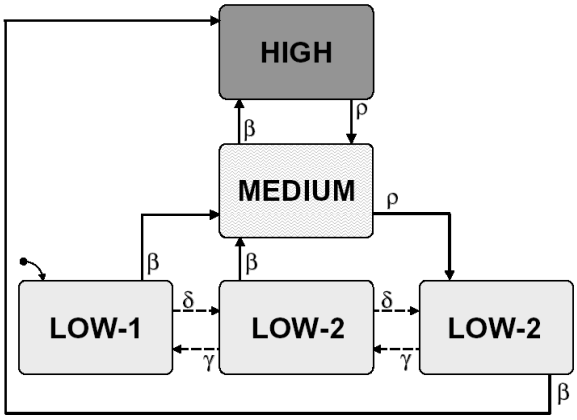

Figure 6. A correct and adequate display

Finally, while a verification methodology such as the one presented here is quite useful for evaluations of interfaces, it is not a panacea for design. For larger and more complex systems, it may take considerable effort to develop and verify one display after another, with no guarantee of success. Furthermore, even when a correct interface is identified, there is no assurance that it is the simplest possible—there could be an equally good, or even better abstraction, hiding just around the corner. The development of a methodology and algorithm for generating interfaces that are both correct and succinct is discussed in a recent NASA report (Heymann & Degani, 2002).

REFERENCES

Abbott, K., Slotte, S. M., & Stimson, D. K. (1996). *The interface between flightcrews and modern flight deck systems*. Washington, DC: Federal Aviation Administration.

Degani, A. & Heymann, M. (2002). Formal Verification of Human-Automation Interaction. *Human Factors, 44 (1)*, 28-43.

Heymann M., & Degani A. (2002). *On abstractions and simplifications in the design of human-automation interfaces*. NASA Technical Memorandum 2002-211397. Moffett Field, CA.

Norman, D. A. (1983). Design rules based on analysis of human error. *Communications of the ACM, 26 (4)*, 254-258.